



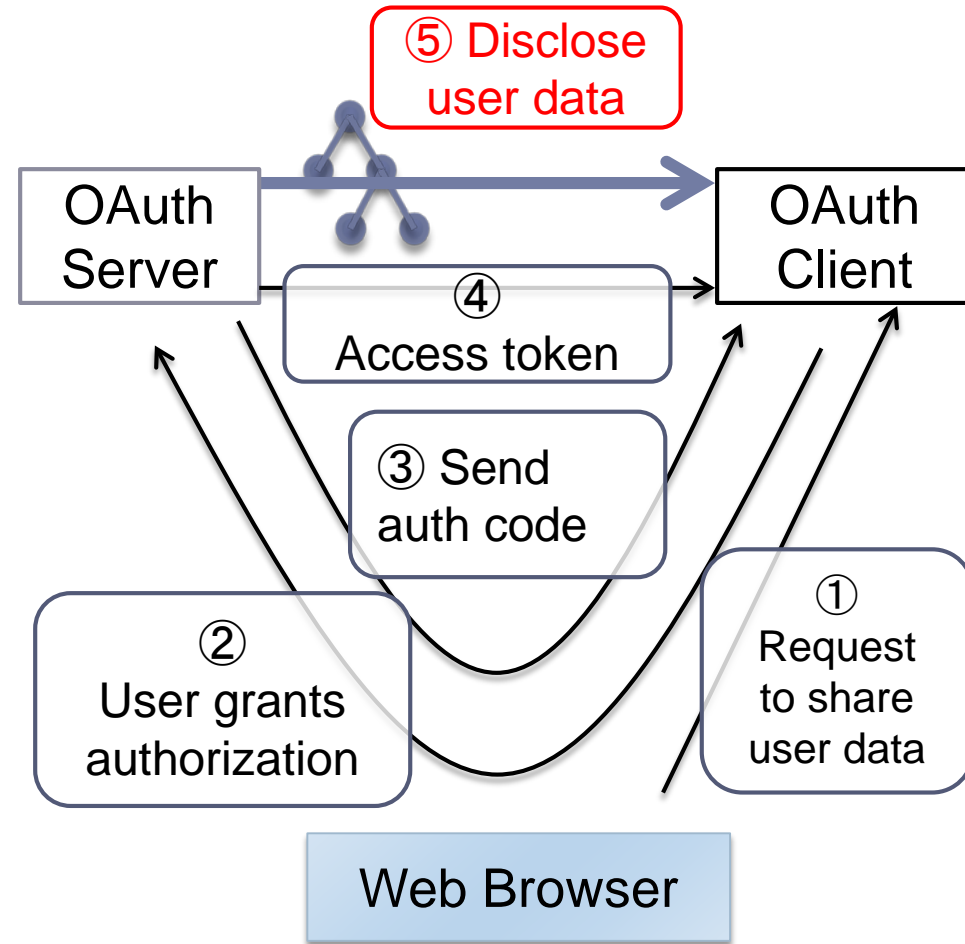
The Web Browser Personalization with the Client Side Triplestore

Canon Inc., W3C/MIT: Hitoshi Uchida



Issue #1 : Silos of Current Web Services

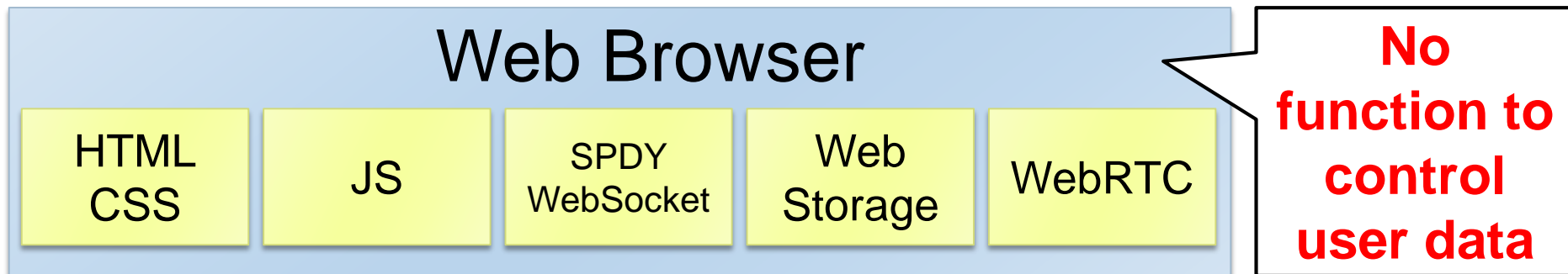
- ▶ Productive web services
 - ▶ All done with web browsers
 - ▶ Document, Presentation, Schedule, Email, Photo, SNS
- ▶ However, not enough options to mash up and utilize user data among services
 - ▶ Standard way only with OAuth
 - ▶ Can't mash up with services which don't support OAuth
 - ▶ Don't pay attention to which user data are still opened
 - ▶ Difficult for general users to understand what happens during redirections



A flow of OAuth protocol

Issue #2 : User Data Centralization on 3rd Parties

- ▶ Functional web browsers with innovative technologies
 - ▶ Rendering rich graphical web pages with CSS
 - ▶ Real-time transfer protocol with SPDY, WebSocket
 - ▶ Video streaming interface with WebRTC
 - ▶ Client data storage with Web Storage, IndexedDB
- ▶ However, we strongly depend on the qualities of 3rd party services to control our personal data on the web
 - ▶ **No** ability to control on web browsers. Just web application execution engine
 - ▶ **Only after** disclosing our private data, we can utilize on the web




What We Want to Achieve

- ▶ Mash up user data without disclosing our private data and personalize browsing experiences with semantic web
 - ▶ **TriplestoreJS** library on HTML5 Web Storage
 - ▶ Collect personal semantic data into web browsers during browsing
 - ▶ Web browser personalization plugin **Semantic Spider**
 - ▶ Enhance user experiences on web browsers with utilizing collected personal semantic data
- } **OSS**
- ▶ Provide new inspirations for HTML5 developers to integrate semantic web



**Semantic Spider
Chrome Extension**

Related Works

- ▶ Knowledge bases on server sides
 - ▶ [DBpedia](#) making tons of knowledge of Wikipedia accessible
 - ▶ [Search engines](#) not only identifying embedded data for web search but also extracting semantic data
 - ▶ Knowledge bases on client sides
 - ▶ Web browser plugins : [Tabulator](#), [Piggy Bank](#)
 - ▶ Easy UI to view/edit collected personal semantic data
 - ▶ Can update semantic data with sending SPARQL update
 - ▶ However, [development tools for experts of SemWeb](#)
 - ▶ General users don't directly get benefits during browsing
 - ▶ Difficult for HTML5 developers not familiar with SemWeb stuffs
-  **We show new potentials for web browsers to provide new user experiences by**

An Architecture of TriplestoreJS

- ▶ Extension on RDFa API supporting save/search/update/delete triples with Web Storage
- ▶ Open source on Github
 - ▶ <https://github.com/shishimaru/triplestoreJS>

HTML5 Application

W3C RDFa API + α

Save

Search

Update

Delete

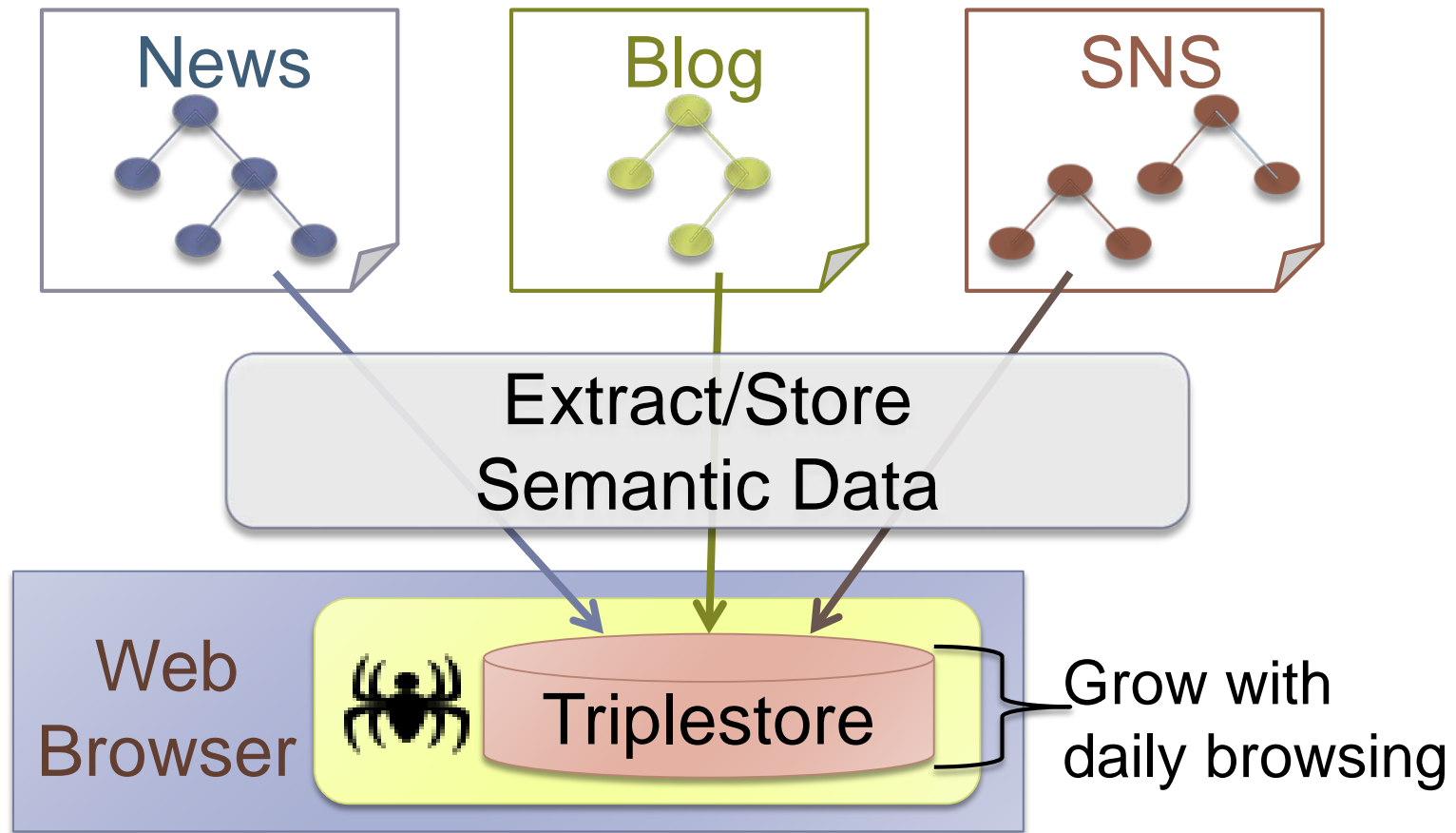
CURIE Process

TriplestoreJS

HTML5 Web Storage

A User Knowledge Base on Web Browser

- ▶ **Semantic Spider Plugin** collects personal semantic data during web browsing
- ▶ Make a personal knowledge base in a web browser



UI for Viewing Personal Semantic Data

- ▶ A sample screenshot on online photo service **Flickr**

The screenshot shows a web browser window with the Flickr website. The address bar displays the URL: `www.flickr.com/photos/31768888@N02/sets/72157642547271133/`. The page title is "San Francisco - a set of". The Flickr navigation bar is visible at the top. A search bar on the left contains the text "San Francisco". A dropdown menu is open, showing a list of item types with their counts:

Types: flickr_photos:set	Count
all	817
article	2
flickr_photos:photostream	1
flickr_photos:set	2
http://schema.org/Comment	420
http://schema.org/Event	288
http://schema.org/ImageObject	34
http://xmlns.com/foaf/0.1/Person	61

Below the dropdown menu, the "Items" section shows a large photo of the Golden Gate Bridge. To the right of the photo, a list of metadata is displayed:

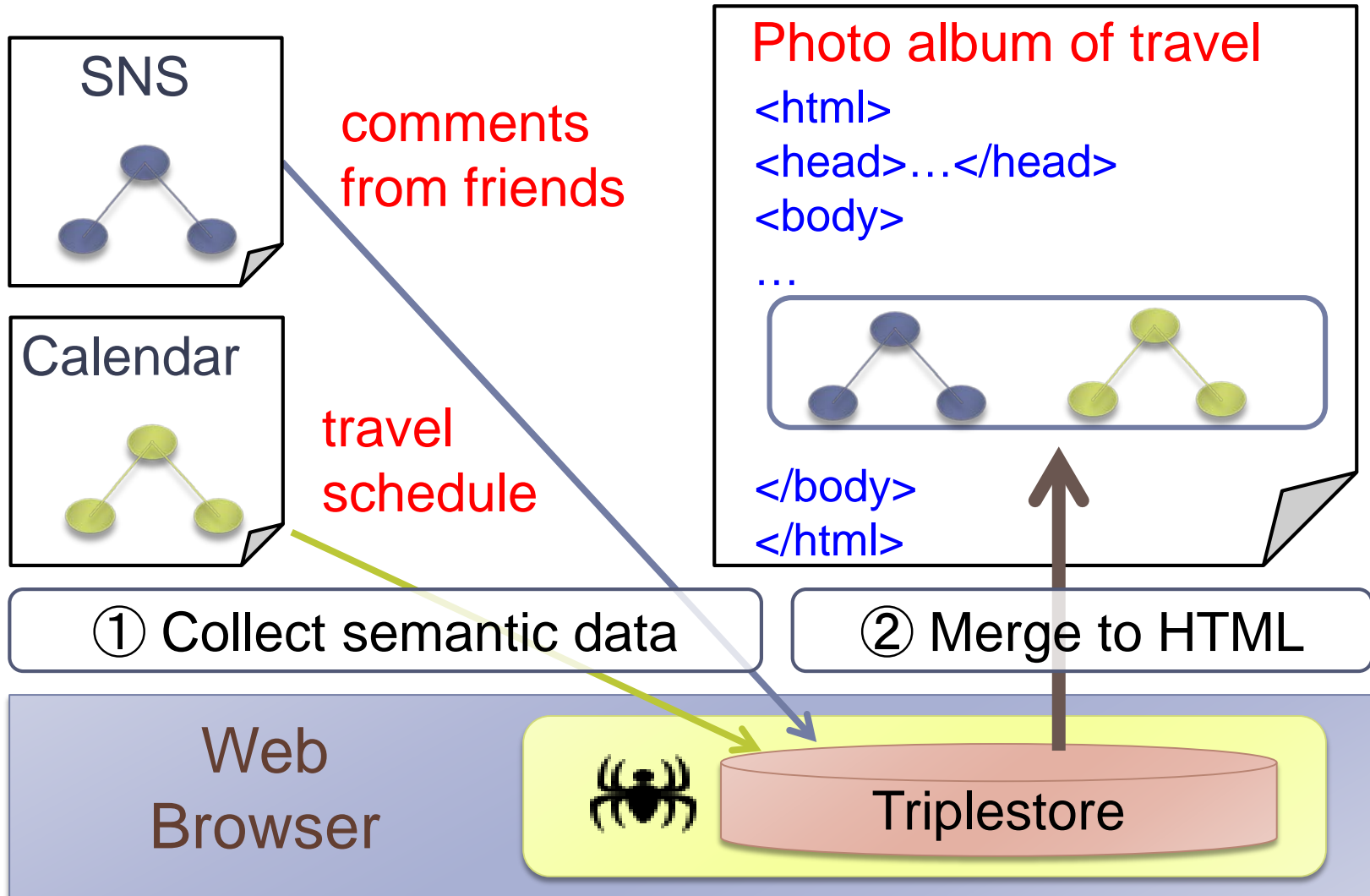
- Alios:app_name : Flickr
- Alios:app_store_id : 328407587
- Fb:app_id : 137206539707334
- Image:height : 425
- Image:width : 640
- Site_name : Flickr
- [Flickr_photos:by](#)
- Twitter:app:id:iphone : 328407587
- Twitter:app:name:iphone : Flickr
- Twitter:card : summary
- Twitter:site : @flickr

Red arrows and brackets on the right side of the image point to specific UI elements:

- Text Search**: Points to the search bar containing "San Francisco".
- Filter with Item type**: Points to the dropdown menu showing item types and counts.
- Search Result**: Points to the large photo of the Golden Gate Bridge and the associated metadata list.

Challenge #1 : Suggesting Related Semantic Data

- Mash up related personal semantic data on any web sites users visit



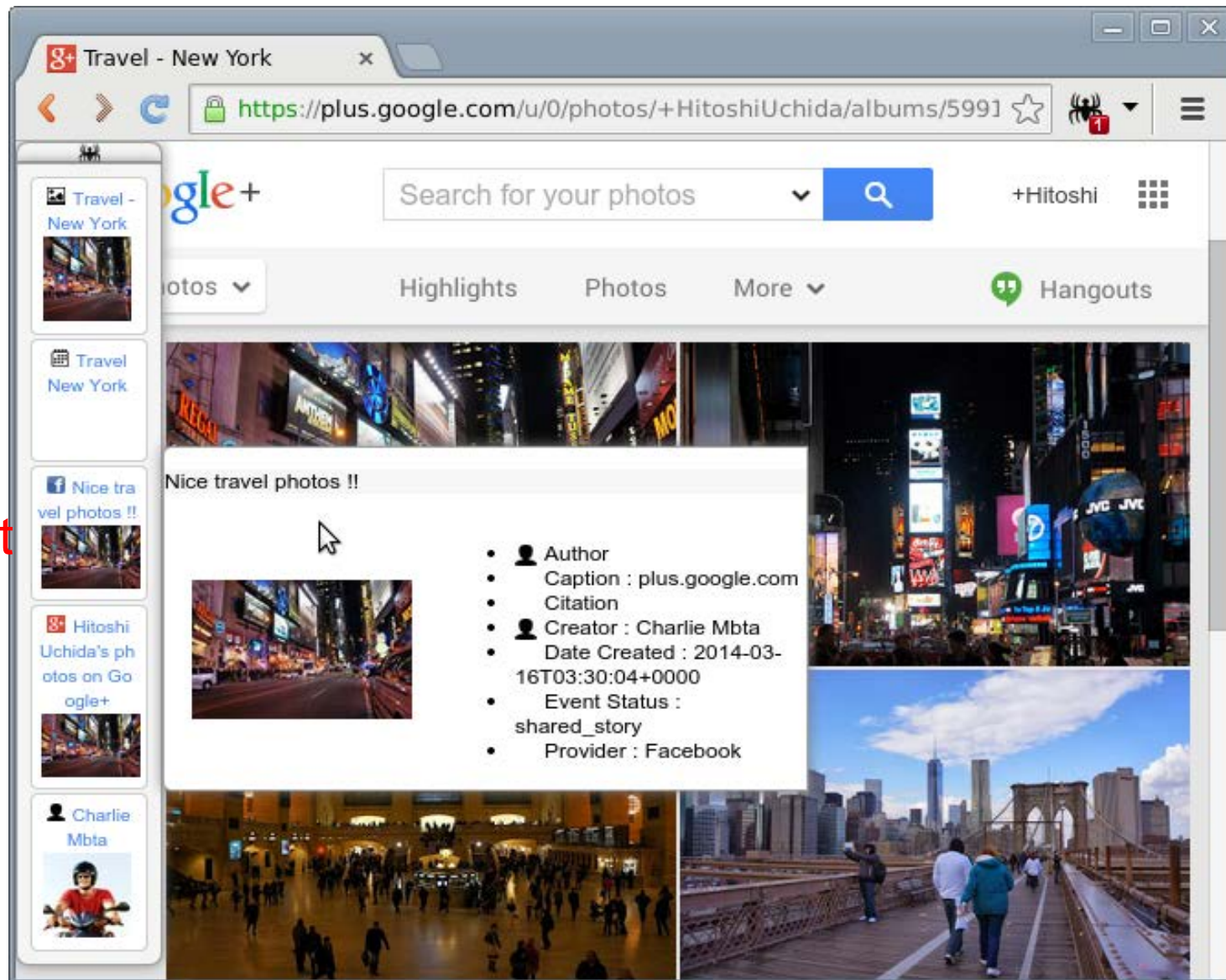
A Screenshot on Google+ Photo Album

Album item

Event Name

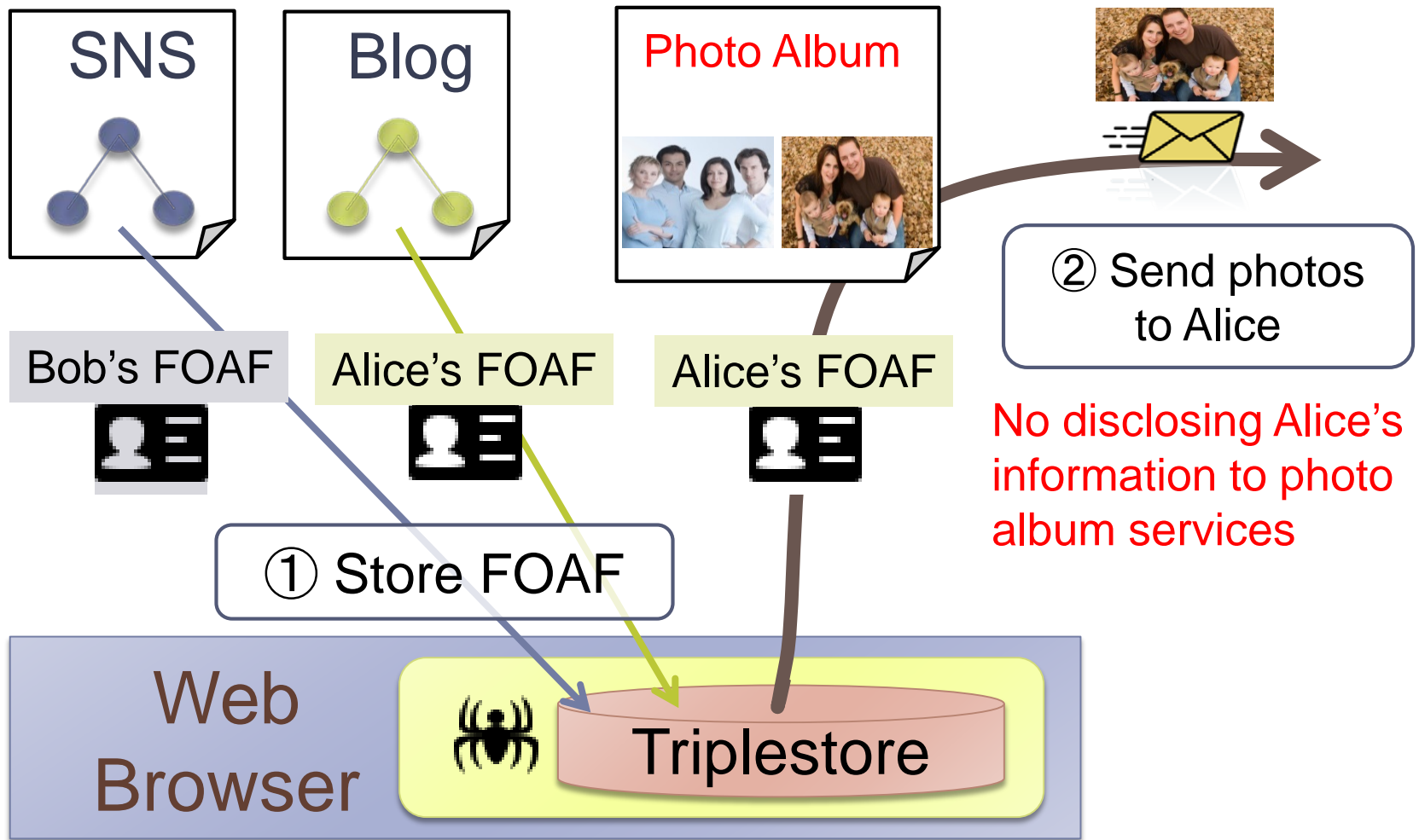
SNS comment

SNS friend
who posted
the comment



Challenge #2 : Assisting Media Sharing Operations

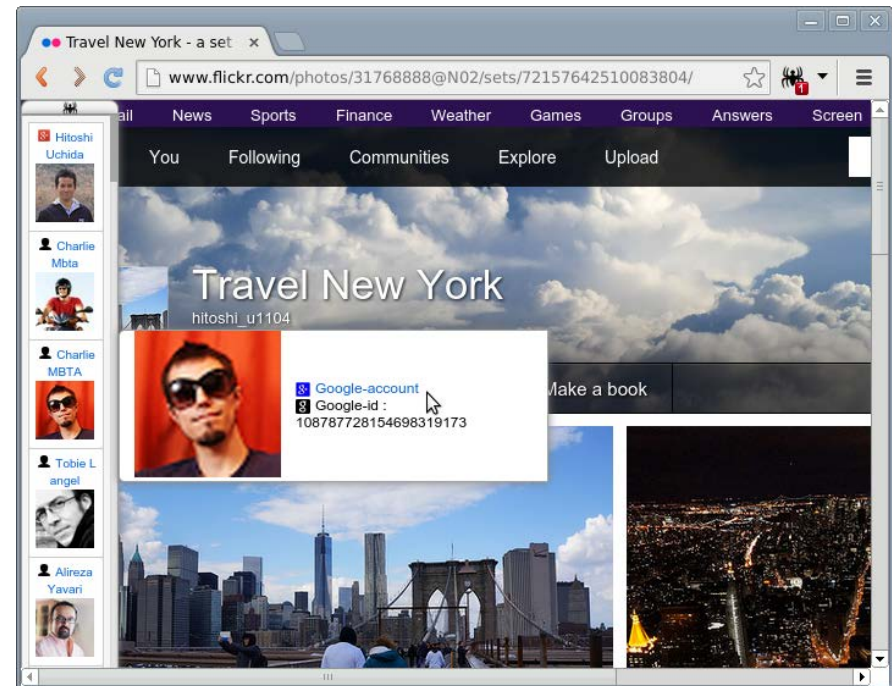
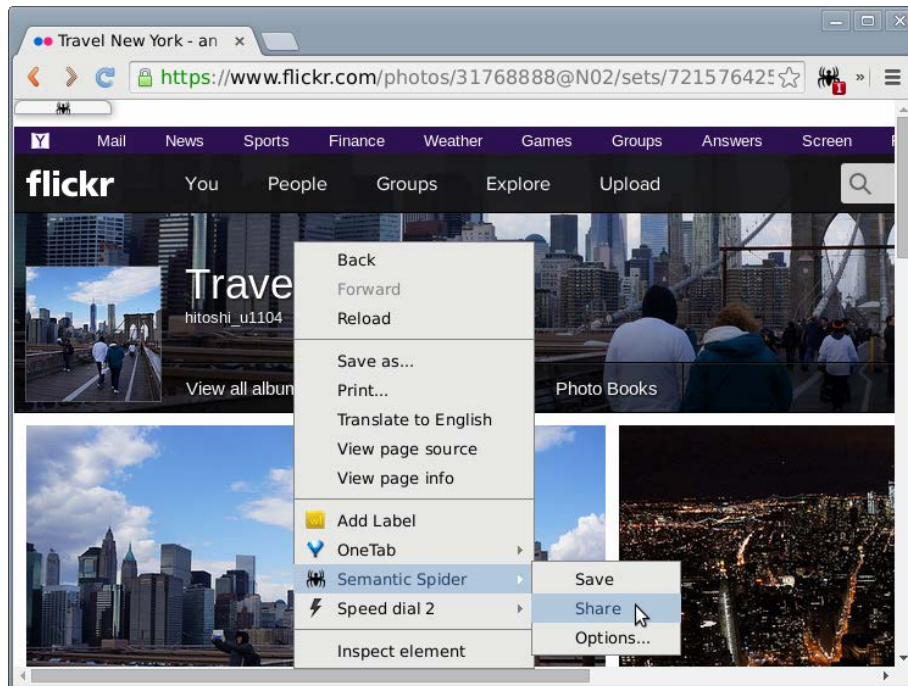
- ▶ Utilize stored FOAF for media sharing with friends via Google+, Facebook, Email



A Screenshot of Suggesting Contact Lists

① Right click a photo and select '**Share**' menu

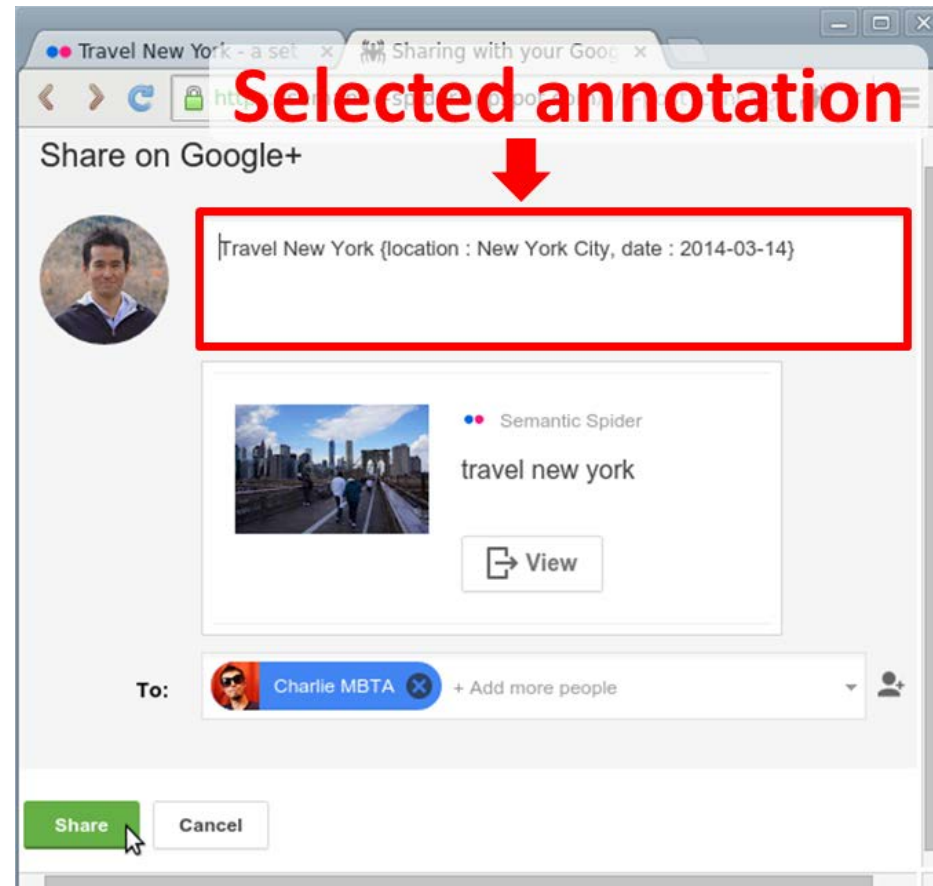
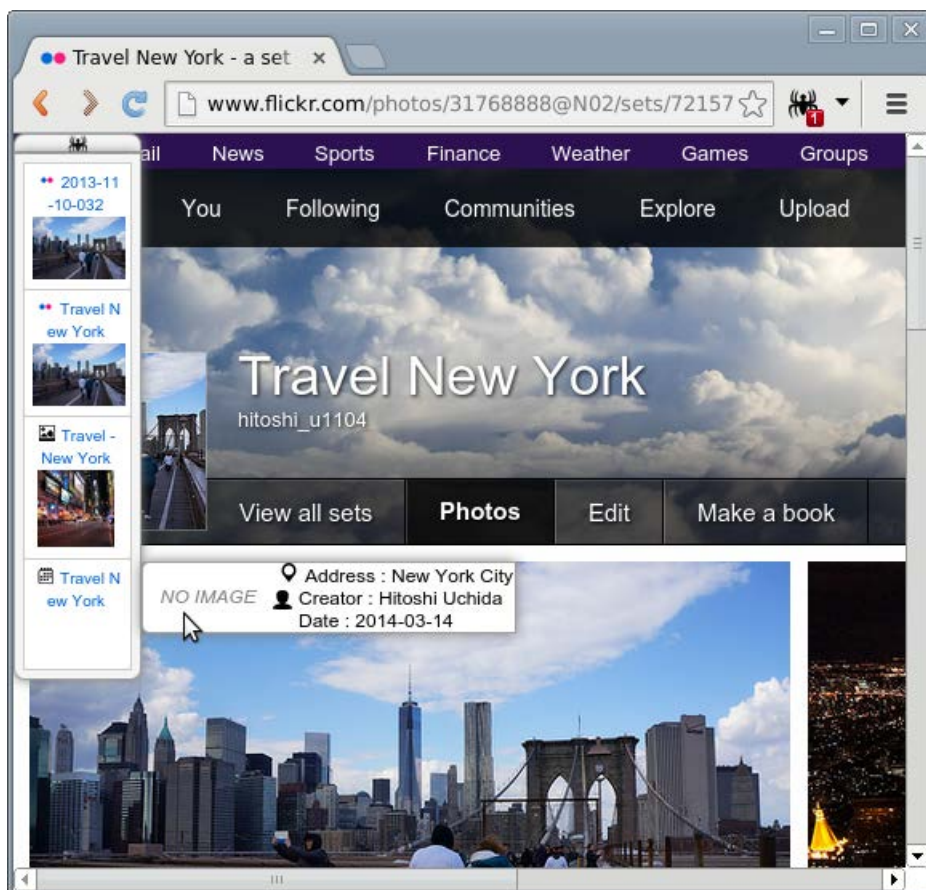
② Select a friend from suggested contact lists



A Screenshot of Sharing an Annotated Photo

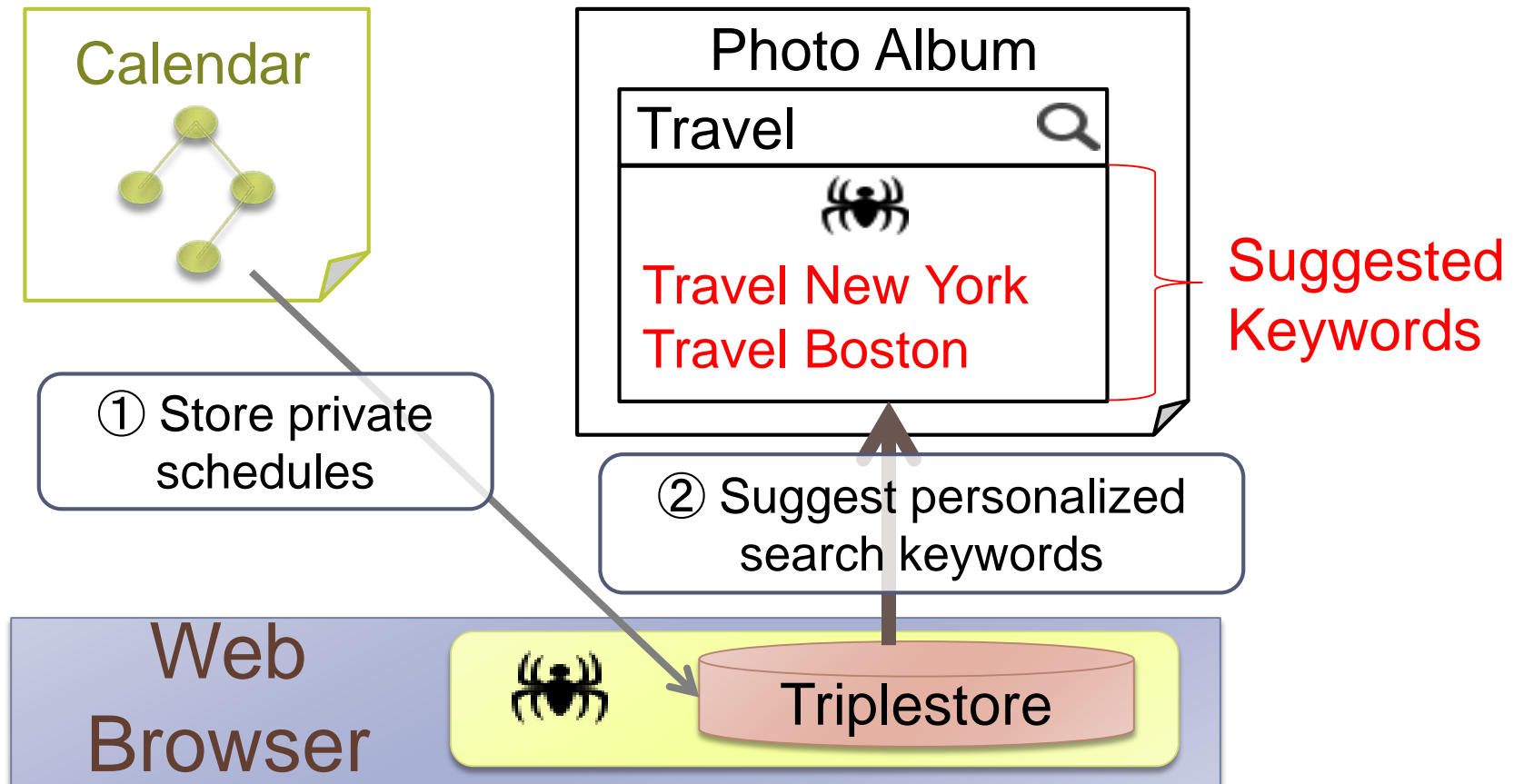
③ Select one of suggested items to **annotate the photo**

④ Share the photo with annotations via Google+



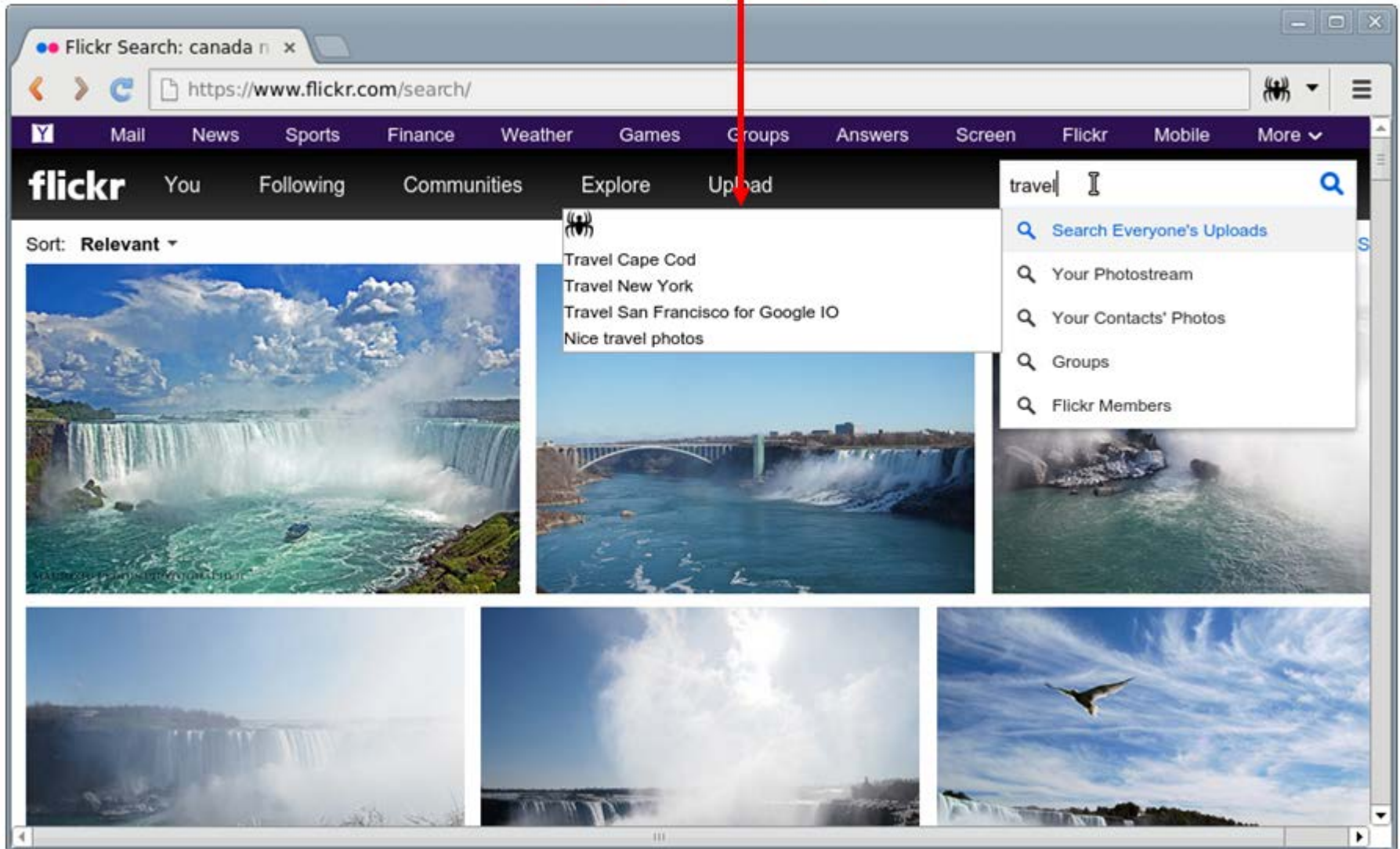
Challenge #3 : Assisting Text Input Operations

- ▶ Utilize personal semantic data to assist text input operations on web pages
- ▶ Support an optional filtering with a item type
 - ▶ `<input type='text' itemtype='http://schema.org/Event'>`



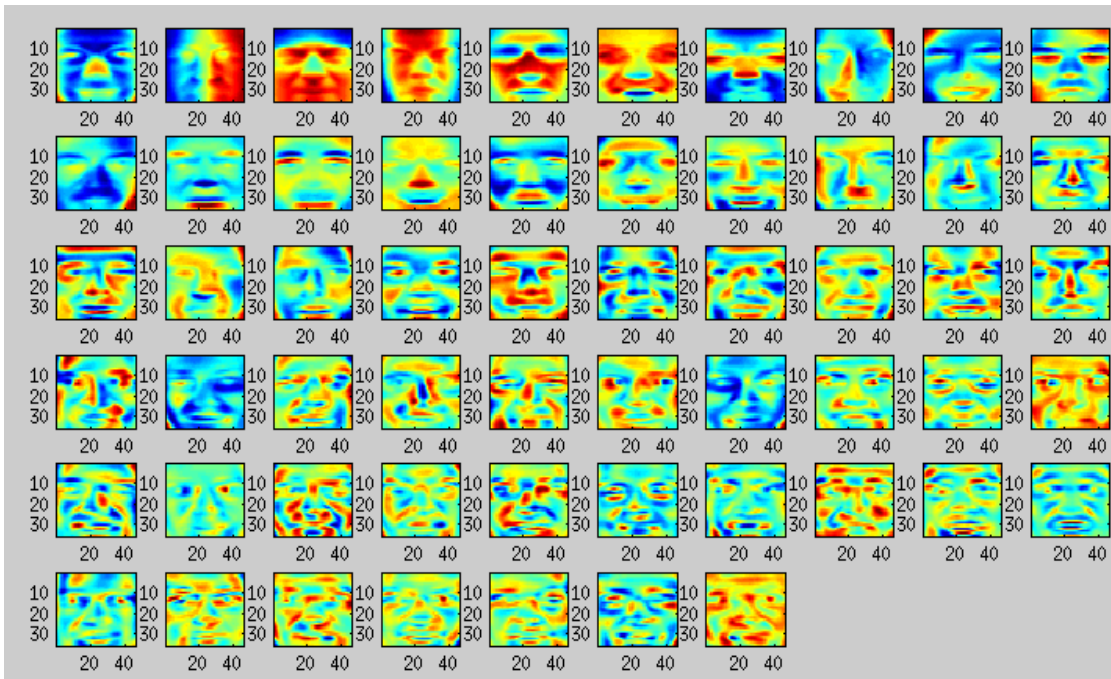
A Screenshot of Keyword Suggestion on Flickr

Suggested keywords



Challenge #4 : Annotating Online Photos with FOAF

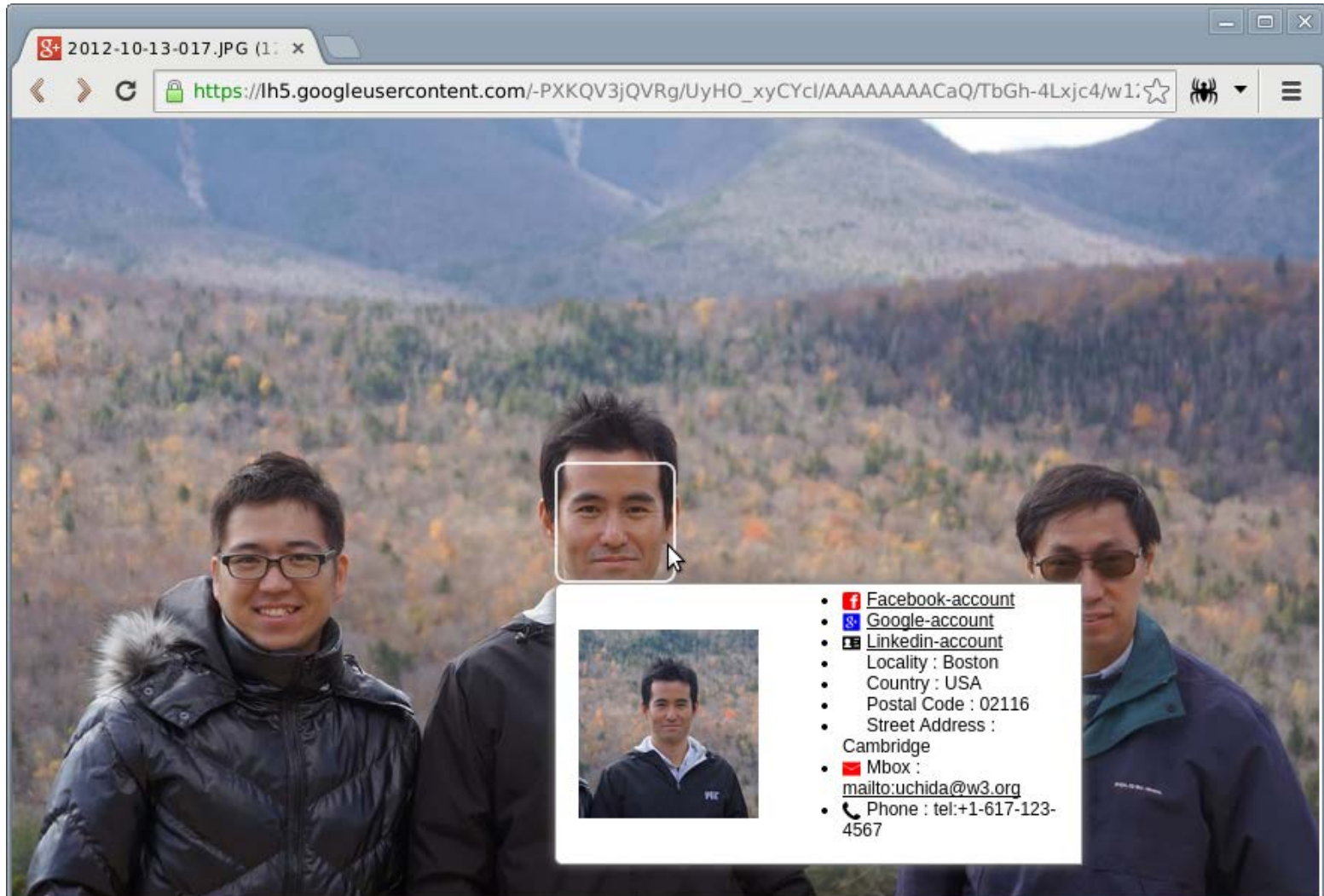
- ▶ Annotate faces of online photos with collected FOAF
 - ▶ No disclosing private information to album services
- ▶ Developed a **face identification using FOAF image data**
 - ▶ Integrated a training result with 13,233 face images as JS codes
 - ▶ Used ccv.js and face.js for face area detection



- Training result Eigenfaces
- Converted into JavaScript to integrate into Semantic Spider plugin

A Screenshot of Face Annotation with FOAF

- ▶ Overlay collected FOAF on Google+ online photos
 - ▶ Users can share the photo with clicking SNS accounts



Conclusion and Future Work

▶ Conclusion

- ▶ Developed **TriplestoreJS** library for HTML5 web applications
- ▶ Developed a web browser personalization plugin, **Semantic Spider**
 - ▶ Utilized personal semantic data collected during web browsing to personalize user experiences

▶ Future work

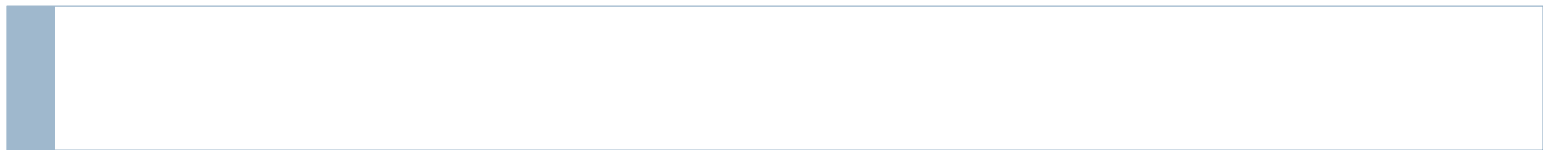
- ▶ Replace underlying Web Storage with **IndexedDB** for better performance
 - ▶ TriplestoreJS can omit conversions from JS object to JSON string
 - ▶ Felt a risk using IndexedDB in W3C Candidate Recommendation when started development
 - ▶ Mobile platforms just started to support IndexedDB
- ▶ **We hope this work will inspire new applications of semantic web and be a promising bridge for HTML5 developers**

Thank you !

- ▶ Demo video
 - ▶ goo.gl/bvCeRN
- ▶ Hitoshi Uchida
 - ▶ Canon Inc.
 - ▶ W3C/MIT



[plug.google.com/+HitoshiUchida](https://plus.google.com/+HitoshiUchida)



A Sample Code with TriplestoreJS

- ▶ Intuitive API for general HTML5 developers

```
var st = new Triplestore();  
//Registers a pair of prefix and URI for CURIE processing  
st.setMapping('foaf', 'http://xmlns.com/foaf/0.1');  
  
//Stores a triple of subject-property-value  
st.add('http://example.org/people#bob', 'foaf:name', 'Bob');  
st.add('http://example.org/people#bob', 'foaf:homepage', 'http://bob.org');  
  
/* Searches properties. Returns  
['http://xmlns.com/foaf/0.1/name', 'http://xmlns.com/foaf/0.1/homepage'] */  
var properties = st.getProperties('http://example.org/people#bob');  
  
//Searches a value. Returns ['Bob']  
var name = st.getValues('http://example.org/people#bob', 'foaf:name');  
  
//Searches all values. Returns ['Bob', 'http://bob.org']  
var values = st.getValues('http://example.org/people#bob', null);
```

Performance Evaluation of TriplestoreJS

- ▶ Linear increase with the number of triples
 - ▶ Used semantic data from web pages, blog, Google+, Facebook

